AF IFW

PATENT
5181-61100/P5026

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.: 09/557,068
Filed: April, 21, 2000
Inventors:
Sai V. Allavarpu, et al.

Title: IDL Event and Request Formatting for CORBA Gateway

§ § § § § § § § § § § § § § §

Examiner: Hu, Jinsong
Group/Art Unit: 2154
Atty. Dkt. No: 5181-61100

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below.

Robert C. Kowert
Printed Name

Signature

November 3, 2005
Date

REPLY BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This response is in reply to the Examiner's Answered dated September 9, 2005. Appellants respectfully request that this Reply Brief be entered pursuant to 37 C.F.R. § 41.41 and considered by the Board of Patent Appeals and Interferences.

REPLY TO EXAMINER'S ANSWER

The Examiner states on page 2 of the Examiner's Answer that the statement of the status of claims contained in Appellants' Appeal Brief is incorrect, but fails to provide any reasons why. Appellants maintain that the status of claims as stated in the Appeal Brief is correct.

First Ground of Rejection:

Claims 1, 3, 5 and 6:

Appellants have argued that Carre does not anticipate a gateway configured to deliver messages between managed objects and one or more managers through a platform-independent interface, wherein the gateway is configurable to deliver the messages for each manager in a format selected by that manager. Instead, Carre pertains to address conversion between CORBA objects and OSI objects (Carre - col. 1, lines 9-19; col. 1, line 59 - col. 2, line 46) and to the transforming of object interfaces (column 5, lines 49-59). Thus, Carre is concerned with converting *address types and object interfaces*, but fails to disclose anything regarding message formats and a gateway that is configurable to deliver the messages for each manager in a format selected by that manager. In fact, Carre specifically teaches the transformation of object interfaces so that a single message format, "classic CORBA messages", may be used with either OSI objects or CORBA objects (Carre, column 5, lines 49-59). Please refer to Appellants' Appeal Brief for a more detailed discussion regarding Carre's failure to anticipate a gateway configurable to deliver the messages for each manager in a format selected by that manager.

In the Examiner's Answer, as well as in the Response to Arguments of the Final Action, the Examiner argues that Carre's system includes multiple gateways and that each communicates with the manager via a different interface. The Examiner further contends that those gateways are thus configured to deliver the message for each manager in a format selected by that manager. The Examiner refers to Carre's teachings regarding

the delivery of messages through different interfaces (e.g. CDMO and CMISE) by gateways and cites Figures 3a and 3b of Carre. The Examiner contends that since Carre teaches more than one gateway and since they each communicate via different interfaces, they perform the same function as a gateway configurable to deliver messages for each manager in a format selected by that manager.

However, the Examiner's interpretation of Carre's interfaces is incorrect. Carre states that his interface units translate an interface to the underlying object so that the underlying object "can be accessed by classic CORBA messages" (Carre, column 5, lines 50-52) (emphasis added). Carre also states that his CMISE/IDL interface appears to the outside like a CORBA object (Carre, column 5, lines 26-31). Specifically, Carre teaches that address conversion is performed *according to the types* of objects that are communicating. Thus, Carre's managers cannot *select* a desired message format. There is clearly no such functionality described in Carre. Instead, managers in Carre only use classic CORBA messages. The sections cited by the Examiner (col. 5, lines 49-59 and col. 6, lines 30-35) refer to address-type conversion between CORBA objects and OSI objects. There is absolutely no mention in Carre of managers being able to select the format for messages delivered by the gateway. Nor does Carre describe any mechanism by which a manager can select a format for messages. Carre fails to mention anything about different message formats. The gateway in Carre is clearly not described as being capable of allowing the managers to select a format.

One portion of Carre cited by the Examiner (column 5, lines 49-59) describes how OSI objects OM and OA can be *transformed into pure CORBA objects* to allow them to be accessed using classic CORBA messages. Thus, Carre is clearly teaching the transformation of object interfaces so that a single message format (i.e. classic CORBA) may be used with either OSI objects or CORBA objects. Furthermore, Carre's manager objects cannot select which interface to communicate through. On the contrary, Carre teaches that his interfaces are present to allow interaction between CORBA and OSI objects "via a CORBA infrastructure" (Carre, column 4, line 63 – column 5, line 3). Carre teaches the use of additional communication layers (GDMO/C++, GDMO/IDL and

CMISE/IDL), or components, between OSI objects and CORBA objects that translate the interfaces to the objects such that they appear as, and are accessible as, CORBA objects *using CORBA messages* (Carre, Figures 2a and 2b, column 5, lines 49-59). In other words, Carre's interfaces are present specifically to provide communication between otherwise incompatible objects using a single message format, "classic CORBA messages".

Furthermore, Carre's interfaces are not message formats as argued by the Examiner. Even if one of Carre's managers could select a different interface, which Appellants maintain they cannot, such a selection would still not be selecting a format for message delivery as the Examiner contends. Additionally, if one of Carre's managers could select a different message format (other than CORBA), that manager would not be able to interact with a target object, which because of Carre's interface transformations is accessible only via classic CORBA messages. Object interfaces and message formats are different things. In Carre's system, different interfaces are provided specifically to allow different object types (specifically OSI or non-CORBA object) to access and send message through a single CORBA infrastructure. Carre's system is quite different from a gateway configured to deliver messages for managers in *formats selected by the managers*.

Furthermore, Carre specifically teaches the use of object interfaces and additional communication layers to allow heterogeneous (CORBA and OSI) objects to communicate via a single, homogeneous infrastructure (CORBA), rather than having a gateway configurable to deliver messages in formats selected by managers. Thus, Carre is teaching away from a gateway configured to deliver messages for managers in formats selected by those managers.

Claim 2:

Appellants have argued in regard to claim 2 that Carre does not anticipate that the selected format comprises text. The Examiner cites column 6, lines 30-35 of Carre.

However, this passage of Carre merely mentions the mapping of ASN.1 onto IDL data types to allow translation of OSI address types to CORBA address types. Carre is not discussing the mapping of ASN.1 to IDL data types or the translation of address types from OSI to CORBA in the context of a format selected by a manager for message delivery. Instead, Carre is talking about translations necessary to allow OSI objects to communicate via classic CORBA. Further, Carre does not teach that such address translations involve a format that comprises text.

In the Examiner's Answer (as well as the Response to Arguments section of the Final Office Action) the Examiner cites the teaching of Carre regarding sending the outcome message to the client based on information required by the client in a request message and further argues that "[a]ll of these messages include context and [are] related to different target object[s]." However, including a request context in a request message does not disclose (or teach or suggest) a client selecting a message format comprising text. A message context may be specified using many different formats other than text. In fact, Carre fails to mention anything about message formats comprising text. Carre simply teaches that request messages can include: an operation, a target object, one or more parameters, and, optionally a request context. The Examiner is merely speculating in hindsight regarding the features of Carre's system. Including a request context in a request message, as taught in Carre, has nothing to do with a manager selecting a delivery format comprising text.

Claims 16, 18, 20 and 21:

Appellants have argued that Carre does not anticipate one or more managers each selecting a format for messages deliverable by a gateway between one or more managed objects and each of the one or more managers, as the examiner contends. As noted above regarding claim 1, Carre pertains to address conversion between CORBA objects and OSI objects (*see,* Carre - col. 1, lines 9-19; col. 1, line 59 - col. 2, line 46) and to the transformation of object interfaces column 5, lines 49-59). Thus, Carre is concerned with

converting *address types and object interfaces*, but fails to disclose anything regarding message formats.

Carre's address conversion is performed according to the types of objects that are communicating. There is no ability in Carre for the managers to select a desired message format. The sections cited by the Examiner (col. 5, lines 49-59 and col. 6, lines 30-35) refer to address-type conversion between CORBA objects and OSI objects. There is absolutely no mention in Carre of managers being able to select the format for messages delivered by the gateway. Nor does Carre does not describe any mechanism by which a manager can select a format for messages. Carre fails to mention anything about different message formats. The gateway in Carre is clearly not capable of allowing the managers to select a format.

In the Examiner's Answer, the Examiner argues that Carre's system includes multiple gateways and that each communicates with the manager via a different interface. The Examiner refers to Carre's teachings regarding the delivery of messages through different interfaces (CDMO and CMISE) by gateways and cites Figures 3a and 3b. The Examiner further contends that because Carre's gateways may use different interfaces, those gateways are configured to deliver the message for each manager in a format selected by that manager. However, the Examiner's interpretation of Carre's interfaces is incorrect. As noted above, Carre does not mention that the manager selects the interface used by a gateway. Specifically, Carre states that his interface units translate an interface to the underlying object so that the underlying object "can be accessed by classic CORBA messages" (Carre, column 5, lines 50-52). Carre also states that his CMISE/IDL interface appears to the outside like a CORBA object (Carre, column 5, lines 26-31).

Please see the arguments above regarding claim 1 for a more detailed discussion of Carre's failure to disclose the ability for managers to select a format for the delivery of messages between managers and managed objects. In summary, Carre teaches only transforming interfaces, which are clearly not messages or message formats. Carre specifically teaches the use of object interfaces and additional communication layers to

allow heterogeneous (CORBA and OSI) objects to communicate via a single, homogeneous infrastructure (CORBA), rather than having a gateway configurable to deliver messages in formats selected by managers. Thus, Carre is teaching away from a gateway configured to deliver messages for managers in formats selected by those managers.

Claim 17:

In regard to claim 17, Carre does not teach that the selected format comprises text. The Examiner cites column 6, lines 30-35 of Carre. However, this passage of Carre merely mentions the mapping of ASN.1 onto IDL data types to allow translation of OSI address types to CORBA address types. Additionally, Carre is not discussing the mapping of ASN.1 to IDL data types or the translation of address types from OSI to CORBA in the context of a format selected by a manager for message delivery. Instead, Carre is talking about the translations necessary to allow OSI objects to communicate via CORBA. Further, Carre does not teach that such address translations involve a format that comprises text.

In the Examiner's Answer (as well as the Response to Arguments section of the Final Office Action) the Examiner cites the teaching of Carre regarding sending the outcome message to the client based on information required by the client in a request message and further argues that "[a]ll of these messages include context and [are] related to different target object[s]." However, as described above regarding claim 2, including a request context in a request message does not disclose (or teach or suggest) a client selecting a message format comprising text. A message context may be specified using many different formats other than text. In fact, Carre fails to mention anything about message formats comprising text. Carre simply teaches that request messages can include: an operation, a target object, one or more parameters, and, optionally a request context. Additionally the arguments presented above regarding claim 2 apply to claim 17 with equal force. Please see the above arguments regarding claim 2 for a more detailed discussion regarding Carre's failure to disclose a selected format comprising text.

Claim 31, 33, 35 and 36:

Regarding claim 31, Carre does not disclose one or more managers each selecting a format for messages deliverable by a gateway between one or more managed objects and each of the one or more managers. As noted above regarding claims 1 and 16, Carre pertains to address conversion between CORBA objects and OSI objects (*see,* Carre - col. 1, lines 9-19; col. 1, line 59 - col. 2, line 46) and to the transformation of object interfaces column 5, lines 49-59). Thus, Carre is concerned with converting *address types and object interfaces*, but fails to disclose anything regarding message formats.

In the Examiner's Answer, the Examiner argues that Carre's system includes multiple gateways and that each communicates with the manager via a different interface. The Examiner further contends that those gateways are thus configured to deliver the message for each manager in a format selected by that manager. The Examiner refers to Carre's teachings regarding the delivery of messages through different interfaces (CDMO and CMISE) by gateways and cites Figures 3a and 3b. The Examiner contends that since Carre teaches more than one gateway and since they each communicate via different interfaces, they perform the same function as a gateway configurable to deliver messages for each manager in a format selected by that manager. However, the Examiner's interpretation of Carre's interfaces is incorrect. Specifically, Carre states that his interface units translate an interface to the underlying object so that the underlying object "can be accessed by classic CORBA messages" (Carre, column 5, lines 50-52). Carre also states that his CMISE/IDL interface appears to the outside like a CORBA object (Carre, column 5, lines 26-31).

Please see the arguments above regarding claims 1 and 16, as well as Appellants Appeal Brief, for a more detailed discussion of Carre's failure to disclose the ability for managers to select a format for the delivery of messages between managers and managed objects. In summary, Carre teaches only transforming interfaces, which are clearly not messages or message formats. Carre specifically teaches the use of object interfaces and

additional communication layers to allow heterogeneous (CORBA and OSI) objects to communicate via a single, homogeneous infrastructure (CORBA), rather than having a gateway configurable to deliver messages in formats selected by managers. Thus, Carre is teaching away from a gateway configured to deliver messages for managers in formats selected by those managers.

Claim 32:

In regard to claim 32, Carre does not teach that the selected format comprises text. The Examiner cites column 6, lines 30-35 of Carre. However, this passage of Carre merely mentions the mapping of ASN.1 onto IDL data types to allow translation of OSI address types to CORBA address types. Additionally, Carre is not discussing the mapping of ASN.1 to IDL data types or the translation of address types from OSI to CORBA in the context of a format selected by a manager for message delivery. Instead, Carre is talking about the translations necessary to allow OSI objects to communicate via CORBA. Further, Carre does not teach that such address translations involve a format that comprises text.

In the Examiner's Answer (as well as the Response to Arguments section of the Final Office Action) the Examiner cites the teaching of Carre regarding sending the outcome message to the client based on information required by the client in a request message and further argues that "[a]ll of these messages include context and [are] related to different target object[s]." However, as described above regarding claim 2, including a request context in a request message does not disclose (or teach or suggest) a client selecting a message format comprising text. A message context may be specified using many different formats other than text. In fact, Carre fails to mention anything about message formats comprising text. Carre simply teaches that request messages can include: an operation, a target object, one or more parameters, and, optionally a request context. Additionally the arguments presented above regarding claims 2 and 17 apply to claim 32 with equal force. Please see the above arguments regarding claims 2 and 17, as

well as Appellants' Appeal Brief, for a more detailed discussion regarding Carre's failure to disclose a selected format comprising text.

Second Ground of Rejection:

Claims 1, 4, 5, 6, 7, 8, 10, 14 and 15:

Appellants have argued that Shank does not anticipate a network management system comprising a gateway configured to deliver messages between managed objects and one or more managers through a platform-independent interface, wherein the gateway is configurable to deliver the messages for each manager in a format selected by that manager. Instead, Shank pertains to providing telephony and media services from a server 110 to an application 140 (Shank, Figure 1, column 1, lines 13-18). **Shank is not concerned with, nor does Shank pertain to, network managers and managed objects.** Instead, Shank describes client-server interfaces. According to Shank, a server may include various service interfaces, such as telephony services 210, media services 220, and basic services 230 that a client may use. Shank's system provides a CORBA ORB 260 for communicating with these interfaces (col. 3, line 31 - col. 4, line 13). As described in Shank, the service interfaces (such as telephony services 210 and media services 220) allow client application 140 to interact with services such as telephone services provided on telephone network 105 and media services provided by various hardware components (col. 7, lines 15-28).

In the Examiner's Answer, as well as in the Response to Arguments of the Final Action, the Examiner refers to Shank's teaching regarding "providing services through media, telephony and basic services interfaces" and further argues that Shank's interfaces perform a message delivery function as a gateway. The Examiner refers to col. 5, lines 39-50 and col. 17, lines 26-37. However, the Examiner's interpretation of Shank is incorrect. These portions of Shank merely give examples of media and telephony services accessible through Shank's interfaces 220 and 210. This portion of Shank has nothing to do with message formats, let alone delivering a message in a format selected by a manager. The Examiner's cited portions have no relevance to managers selecting

message formats. The Player, Recognizer, etc. discussed in Shank are media services, not managers for managed objects in a managed network. Moreover, there is clearly no teaching in Shank of a manager using any these services to select a format for message delivery. Shank's interfaces are *defined according to the target object or target hardware*, such as text-to-speech services 222, or facsimile services 228, and the *formats of messages are not selected by a manager managing* such a target object. Furthermore, as with the rejection of claim 1 over Carre, the Examiner seems to be confusing interfaces with message formats. Different interfaces do not imply selectable delivery formats. Appellants further point out that the Examiner's cited portions of Shank (column 5, lines 39-50 and column 17, lines 26-37) teach only that different interfaces may include different *method* definitions, but fail to teach anything regarding the format of messages and further fail to teach *message formats selectable by a manager.*

Contrary to the Examiner's assertions, the service interfaces 210, 220 and 230 of Shank's server 110, do not provide a gateway configurable to deliver the messages for each manager in a format selected by that manager. Shank does not pertain to interactions between managers and managed objects as these entities are understood in the art. Instead, Shank only discusses the client-server interactions between application 140 and server 110. In other words, Shank only discusses providing telephony and media services through a server to a client application. As discussed above, Shank's interfaces 210, 220, 230 provide service interfaces for an application 140. They do not deliver messages between managed objects and one or more managers. Telephony service interface 210 is not a manager for managed objects. The concept of managers and managed objects, and the relationship between managers and managed objects, is well understood in the art of managed networks. In contrast, telephony service interface 210 (including 212-216) is clearly described in Shank as providing an interface for application 140 to access services on telephony network 105. Thus, interfaces 210-216 in Shank are not described as having anything to do with managing managed objects on a managed network.

Additionally, Shank teaches that his service interfaces are defined according to the target object or target hardware, such as text-to-speech services 222, or facsimile services 228, and fails to teach that the formats of messages are selected by a manager managing such a target object. In fact, Shank teaches the use of a custom format "based on similar methods specified in the ECTF S 1.00API," but defined using IDL (Shank, column 17, lines 31-34). Data used by these interfaces is "in the form of a key value set (KVS) which contains a sequence of keys associated with values " and "[s]tructurally, a KVS is a sequence of key value pairs (KVPairs)" (Shank, column 9, lines 1-7). According to Shank, application 140, which the Examiner has erroneously characterized as a manager, communicates with various services using whatever interface the service has registered with resource administrator 236 (Shank, column 5, lines 16-22). Shank does not teach that application 140 selects a message format when communicating with services. Instead, Shank clearly teaches the use of predefined message formats and not the use of formats selectable by a manager.

Claim 2:

Appellants have argued in regard to claim 2 that Shank does not disclose wherein the selected format comprises text. The Examiner cites item 228 of Figure 2. However, item 228 refers to a FAX service, and it is well known that a facsimile interface does not include text, but rather involves a graphic interface. Appellants fail to see the relevance of item 228 of Figure 2 to a selected format that comprises *text*.

In the Examiner's Answer, the Examiner cites Shank's teachings regarding providing text-to-speech services. The text-to-speech services referred to by the Examiner are services accessed by client application 140. For example, the text-to-speech service converts text data supplied by application 140 into an audio file. When discussing text-to-speech, Shank is referring to a high level function performed by the service, not an inter-object message format used for communicating with the service. Even if Shank's application 140 were to represent a manager object, which Appellants assert it does not, Shank still does not teach that application 140 may choose text-to-

speech as a format for message delivery when communicating with a service object. In fact, it is clear that when communicating with a text-to-speech service, one must provide the text to be converted into audio. Shank teaches that application 140 must use an interface specified by the text-to-speech service's ORB vendor (Shank, column 3, line 65-column 4, line 6). Thus, Shank's text-to-speech service is not a selectable message format for communicating between a manager and a managed object. Shank clearly does not describe a manager being able to select a format for message delivery that comprises text.

Claim 10:

In regard to claim 10, Shank does not anticipate a gateway comprising a request gateway which is configured to deliver messages generated by the one or more managers to the one or more managed objects, wherein the messages comprises requests and wherein the requests comprise a query for information concerning one of the managed objects. The portions of Shank cited by the Examiner (column 2, lines 64-67; column 7, lines 43-46; and column 7, line 66 through column 8, line 6) refer to application 140 invoking functions of the telephony and media services. These teachings have nothing to do with a query for information concerning a managed object. The concepts of managers and managed objects are well understood in the art of managed networks. Managers and managed objects have a well-known relationship in managed networks. Shank does not pertain to interactions between managers and managed objects, as these entities are understood in the art. In contrast, Shank only discusses the client-server interactions between application 140 and server 110. In other words, Shank only discusses providing telephony and media services through a server to a client application. Shank does not discuss managing managed network objects. Furthermore, nowhere does Shank teach a query for information concerning a managed object.

In the Examiner's Answer, the Examiner cites item 230 of FIG. 2 and column 5, lines 13-26 of Shank and argues that Shank discloses requesting the available resources by the user based on the interface type and properties. However, the Examiner's

characterization of the cited passage is incorrect. The cited passage describes Shank's resource administrator and how resources may register with the resource administrator by providing information including type and, optionally, any attributes or properties that distinguish it from other resources. Shank states that resources "can query resource administrator 236 for the existence of resources by interface type and an optional list of properties" (Shank, column 5, lines 20-22). Thus, the cited passage does not disclose a user requesting available resources, as the Examiner asserts. Instead, it teaches that resources may query a resource administrator regarding other resources. However, a resource making a query regarding other resources is not equivalent to a message generated by a manager to a managed object where the message includes a query for information concerning a managed object. Shank's resources are clearly not managers, and Shank's resource administrator is not a managed object. Thus, any query a resource makes to the resource administrator cannot be considered a message generated by a manager to a managed object. The Examiner has clearly misinterpreted the teachings of Shank.

Claim 11:

Regarding claim 11, Shank does not disclose wherein the requests comprise a command to set one or more parameters of one of the managed objects. The Examiner cites column 17, lines 53-66 where Shank describes parameters for a specific function (Play) of the Player media service interface, but does not mention a command to set parameters for a managed object. A parameter to a specific service method invocation is very different from a command to set one or more parameters of a managed object and Shank does not mention such a command, just that parameters may be used when invoking specific service method invocations.

In the Examiner's Answer, the Examiner cites column 10, lines 55-61 of Shank and refers to Shank's setParametersAsync used to set session parameters. The Examiner further states, "calling the objects satisfy the request criteria." However, the setParameterAsync method is a part of Shank's Session IDL API that "provides a logical

binding between application 140 and media server 200" (Shank, column 8, lines 37-40). The setParameterAsync method is not a message generated by a manager to a managed object, but instead is between an application and a media server. Additionally, Shank's setParameterAsync method does not set parameters of a managed object. Instead, setParameterAsync sets session parameters, which are not parameters of a data resource, which the Examiner equates to the managed objects of Appellants' claims.

Claim 13:

In regard to claim 13, Shank fails to anticipate, wherein requests are converted from the interface definition language to a platform-specific format prior to delivery to the managed objects. The Examiner refers to col. 5, lines 39-50, of Shank. The Examiner's cited portion of Shank discusses examples of media and telephony services but teaches nothing about converting requests from the interface definition language to a platform-specific format prior to delivery to the managed objects. In fact, nowhere does Shank teach or even suggest that requests are converted. Thus, Appellants can see no basis for the Examiner's contention regarding converting requests and assert that the Examiner is merely speculating about Shank's system including converting requests.

In the Examiner's Answer, as well as in the Response to Arguments section of the Final Action, the Examiner refers to Shank's teachings regarding the communication with different objects by different protocols "based on an industry standard" and further argues, "ASN1 can be implemented in Shank's system." Appellants fail to find any relevance to the Examiner's reference to ASN1. Shank does not mention ASN1 at all and certainly does not teach or suggest the use of ASN1. Furthermore, even if ASN1 were to be implemented in Shank's system, that would not require that Shank's system include the translation of request messages. The Examiner seems to be implying that it would be obvious to modify Shank's system to include the translation of request messages; however, a rejection based on such modification is clearly improper in a § 102(e) anticipation rejection.

The Examiner further argues that Shank teaches "converting requests before delivering them to objects when the user and server [execute] in different process[es]" and cites column 4, lines 35-40 of Shank. However, the portion of Shank cited by the Examiner discusses the marshalling of a method invocation in order to communicate between a client in one process and a server in a different process. Marshalling of parameters for method invocations does not involve the translation of any request messages. Shank is not discussing translating request messages at all, but rather Shank is discussing parameters used in inter-process communication. Shank clearly does not anticipate that requests are converted from an interface definition language to a platform-specific format prior to delivery to the managed objects.

The Examiner also contends that the format of converted requests in Shank would be based on the particular receiver, and thus, "they could be any industry standard format, such as Portable Management Interface (PMI) format." As noted above, the Examiner is attempting to reject claim 13 based on features that might or could be implemented in Shank's system, not based upon what is actually disclosed by Shank, which is clearly improper in a rejection based on anticipation (i.e. § 102).

Claim 16, 19, 20, 21, 22, 23, 24, 29 and 30:

Appellants have argued that Shank does not anticipate one or more managers each selecting a format for messages deliverable by a gateway between one or more managed objects and each of the one or more managers. Instead, Shank pertains to providing telephony and media services from a server 110 to an application 140 (Shank, Figure 1, column 1, lines 13-18). Shank is not concerned with, nor does Shank pertain to, managers and managed objects. According to Shank, a server may include various service interfaces, such as telephony services 210, media services 220, and basic services 230 that a client may use. Shank's system provides a CORBA ORB 260 for communicating with these interfaces (col. 3, line 31 - col. 4, line 13). As described in Shank, the service interfaces (such as telephony services 210 and media services 220) allow client application 140 to interact with services such as telephone services provided

on telephone network 105 and media services provided by various hardware components (col. 7, lines 15-28). Please refer to the discussion of claim 1 above for a more detailed discussion regarding Shank's failure to anticipate managers each selecting a format for messages deliverable by a gateway between managed objects and each of the managers.

In the Examiner's Answer, as well as in the Response to Arguments of the Final Action, the Examiner refers to Shank's teaching regarding "providing services through media, telephony and basic services interfaces" and further argues that Shank's interfaces perform a message delivery function as a gateway. However, the Examiner's interpretation of Shank is incorrect. As described above, Shank's interfaces are *defined according to the target object or target hardware*, such as text-to-speech services 222, or facsimile services 228, and the *formats of messages are not selected by a manager managing* such a target object. Furthermore, as with the rejection of claim 1 over Carre, the Examiner seems to be confusing interfaces with message formats. Different interfaces do not imply selectable delivery formats. Appellants further point out that the Examiner's cited portions of Shank (column 5, lines 39-50 and column 17, lines 26-37) teach only that different interfaces may include different *method* definitions, but fail to teach anything regarding the format of messages and further fail to teach *message formats selectable by a manager.*

The Examiner appears to be arguing that by merely using different formats to communicate with different devices, the format must necessarily be selected by the receiver. However, just because Shank may send one format to a telephony device and another format to a facsimile service, does not imply that the telephony device or the software service for the telephony device (or the facsimile device) selects the format. Having a receiver select a format is very different from a sender using a different format.

Please refer to the arguments above regarding claim 1 for a more detailed discussion of Shank's failure to disclose managers selected a desired message format. In short, Shank only discusses providing telephony and media services through a server to a client application. Shank's interfaces 210, 220, 230 provide service interfaces for an

application 140. They do not deliver messages between managed objects and one or more managers. Shank clearly does not teach a gateway that is configurable to deliver the messages for each manager in a format selected by that manager. Shank teaches the use of a custom format "based on similar methods specified in the ECTF S 1.00API," but defined using IDL (Shank, column 17, lines 31-34). Thus, Shank clearly teaches the use of predefined message formats and not the use of formats selectable by a manager.

Claim 17:

In regard to claim 17, Shank does not disclose wherein the selected format comprises text, as expressed by the Examiner. The Examiner cites item 228 of Figure 2. However, item 228 refers to a FAX service, and it is well known that a facsimile interface does not include text, but rather involves a graphic interface. Appellants fail to see the relevance of item 228 of Figure 2 to a selected format that comprises *text*.

In the Examiner's Answer, as well as in the Response to Arguments of the Final Action, the Examiner cites Shank's teachings regarding providing text-to-speech services. The text-to-speech services referred to by the Examiner are services accessed by client application 140. For example, the text-to-speech service converts text data supplied by application 140 into an audio file. When discussing text-to-speech, Shank is referring to a high level function performed by the service, not an inter-object message format used for communicating with the service. Even if Shank's application 140 were to represent a manager object, which Appellants assert it does not, Shank still does not teach that application 140 may choose text-to-speech as a format for message delivery when communicating with a service object. In fact, it is clear that when communicating with a text-to-speech service, one must provide the text to be converted into audio. Shank teaches that application 140 must use an interface specified by the text-to-speech service's ORB vendor (Shank, column 3, line 65-column 4, line 6). Thus, Shank's text-to-speech service is not a selectable message format for communicating between a manager and a managed object. Shank clearly does not describe a manager being able to select a format for message delivery that comprises text. Additionally, please refer to the

discussion above regarding claim 2 for a more detailed discussion regarding Shank's failure to disclose a manager being able to select a format for message delivery that comprises text.

Claim 25:

In regard to claim 25, Shank does not anticipate a gateway comprising a request gateway which is configured to deliver messages generated by the one or more managers to the one or more managed objects, wherein the messages comprises requests and wherein the requests comprise a query for information concerning one of the managed objects. The portions of Shank cited by the Examiner (column 2, lines 64-67; column 7, lines 43-46; and column 7, line 66 through column 8, line 6) refer to application 140 invoking functions of the telephony and media services. These teachings have nothing to do with a query for information concerning a managed object. The concepts of managers and managed objects are well understood in the art of managed networks. Managers and managed objects have a well-known relationship in managed networks. Shank does not pertain to interactions between managers and managed objects, as these entities are understood in the art. In contrast, Shank only discusses the client-server interactions between application 140 and server 110. In other words, Shank only discusses providing telephony and media services through a server to a client application. Shank does not discuss managing managed network objects. Furthermore, nowhere does Shank teach a query for information concerning a managed object.

In the Examiner's Answer the Examiner cites item 230 of FIG. 2 and column 5, lines 13-26 of Shank and argues that Shank discloses requesting the available resources by the user based on the interface type and properties. However, the Examiner's characterization of the cited passage is incorrect. The cited passage describes Shank's resource administrator and how resources may register with the resource administrator by providing information including type and, optionally, any attributes or properties that distinguish it from other resources. Shank states that resources "can query resource administrator 236 for the existence of resources by interface type and an optional list of

properties" (Shank, column 5, lines 20-22). Thus, the cited passage does not disclose a user requesting available resources, as the Examiner asserts. Instead, it teaches that resources may query a resource administrator regarding other resources. However, a resource making a query regarding other resources is not equivalent to a message generated by a manager to a managed object where the message includes a request that comprises a query for information concerning a managed object. Shank's resources are clearly not managers, and Shank's resource administrator is not a managed object. Thus any query a resource makes to the resource administrator cannot be considered a message generated by a manager to a managed object.

Claim 26:

Regarding claim 26, Shank does not disclose wherein the requests comprise a command to set one or more parameters of one of the managed objects. The Examiner cites column 17, lines 53-66 where Shank describes parameters for a specific function (Play) of the Player media service interface, but does not mention a command to set parameters for a managed object. A parameter to a specific service method invocation is very different from a command to set one or more parameters of a managed object and Shank does not mention such a command, just that parameters may be used when invoking specific service method invocations.

In the Examiner's Answer the Examiner cites column 10, lines 55-61 of Shank and refers to Shank's setParametersAsync used to set session parameters. The Examiner further states, "calling the objects satisfy the request criteria." However, the setParameterAsync method is a part of Shank's Session IDL API that "provides a logical binding between application 140 and media server 200" (Shank, column 8, lines 37-40). The setParameterAsync method is not a message generated by a manager to a managed object, but instead is between an application and a media server. Additionally, Shank's setParameterAsync method does not set parameters of a managed object. Instead, setParameterAsync sets session parameters, which are not parameters of a data resource, which the Examiner equates to the managed objects of Appellants' claims.

Claim 28:

In regard to claim 28, Appellants have argued that Shank fails to anticipate, wherein requests are converted from the interface definition language to a platform-specific format prior to delivery to the managed objects. The Examiner refers to col. 5, lines 39-50, of Shank. The Examiner's cited portion of Shank discusses examples of media and telephony services but teaches nothing about converting requests from the interface definition language to a platform-specific format prior to delivery to the managed objects. In fact, nowhere does Shank teach or even suggest that requests are converted. Thus, appellants can see no basis for the Examiner's contention regarding converting requests and assert that the Examiner is merely speculating about Shank's system including converting requests.

In the Examiner's Answer, the Examiner refers to Shank's teachings regarding the communication with different objects by different protocols "based on an industry standard" and further argues, "ASN1 can be implemented in Shank's system." Appellants fail to find any relevance to the Examiner's reference to ASN1. Shank does not mention ASN1 at all and certainly does not teach or suggest the use of ASN1. Furthermore, even if ASN1 were to be implemented in Shank's system, that would not require that Shank's system include the translation of request messages. The Examiner seems to be implying that it would be obvious to modify Shank's system to include the translation of request messages; however, a rejection based on such modification is clearly improper in a § 102(e) anticipation rejection.

The Examiner further argues that Shank teaches "converting requests before delivering them to objects when the user and server [execute] in different process[es]" and cites column 4, lines 35-40 of Shank. However, this portion of Shank discusses the marshalling of a method invocation in order to communicate between a client in one process and a server in a different process. Marshalling of parameters for method invocations does not involve the translation of any request messages. Shank is not

discussing translating request messages at all, but rather Shank is discussing parameters used in inter-process communication.

Shank clearly does not anticipate wherein requests are converted from an interface definition language to a platform-specific format prior to delivery to the managed objects.

Claim 31, 34, 35, 36, 37, 38, 39, 44 and 45:

Shank does not anticipate one or more managers each selecting a format for messages deliverable by a gateway between one or more managed objects and each of the one or more managers. Instead, Shank pertains to providing telephony and media services from a server 110 to an application 140 (Shank, Figure 1, column 1, lines 13-18). Shank is not concerned with, nor does Shank pertain to, managers and managed objects. According to Shank, a server may include various service interfaces, such as telephony services 210, media services 220, and basic services 230 that a client may use. Shank's system provides a CORBA ORB 260 for communicating with these interfaces (col. 3, line 31 - col. 4, line 13). As described in Shank, the service interfaces (such as telephony services 210 and media services 220) allow client application 140 to interact with services such as telephone services provided on telephone network 105 and media services provided by various hardware components (col. 7, lines 15-28).

In the Examiner's Answer, the Examiner refers to Shank's teaching regarding "providing services through media, telephony and basic services interfaces" and further argues that Shank's interfaces perform a message delivery function as a gateway. However, the Examiner's interpretation of Shank is incorrect. As described above, Shank's interfaces are *defined according to the target object or target hardware*, such as text-to-speech services 222, or facsimile services 228, and the *formats of messages are not selected by a manager managing* such a target object. Furthermore, as with the rejection of claim 1 over Carre, the Examiner seems to be confusing interfaces with message formats. Different interfaces do not imply selectable delivery formats.

Appellants further point out that the Examiner's cited portions of Shank (column 5, lines 39-50 and column 17, lines 26-37) teach only that different interfaces may include different *method* definitions, but fail to teach anything regarding the format of messages and further fail to teach *message formats selectable by a manager*.

As noted above regarding claims 1 and 16, the Examiner appears to be arguing that by merely using different formats to communicate with different devices, the format must necessarily be selected by the receiver. However, just because Shank may send one format to a telephony device and another format to a facsimile service, does not imply that the telephony device or the software service for the telephony device (or the facsimile device) selects the format. Having a receiver select a format is very different from a sender using a different format. Additionally, please refer to the arguments above regarding claims 1 and 16 for a more detailed discussion of Shank's failure to disclose managers selected a desired message format.

Claim 32:

In regard to claim 32, Shank does not disclose wherein the selected format comprises text, as expressed by the Examiner. The Examiner cites item 228 of Figure 2. However, item 228 refers to a FAX service, and it is well known that a facsimile interface does not include text, but rather involves a graphic interface. Appellants fail to see the relevance of item 228 of Figure 2 to a selected format that comprises *text*. Please refer to the discussion of claims 2 and 17 for a more detailed discussion regarding Shank's failure to disclose a manager being able to select a format for message delivery that comprises text.

In the Examiner's Answer, as well as in the Response to Arguments of the Final Action, the Examiner cites Shank's teachings regarding providing text-to-speech services. The text-to-speech services referred to by the Examiner are services accessed by client application 140. For example, the text-to-speech service converts text data supplied by application 140 into an audio file. When discussing text-to-speech, Shank is

referring to a high level function performed by the service, not an inter-object message format used for communicating with the service. Even if Shank's application 140 were to represent a manager object, which Appellants assert it does not, Shank still does not teach that application 140 may choose text-to-speech as a format for message delivery when communicating with a service object. In fact, it is clear that when communicating with a text-to-speech service, one must provide the text to be converted into audio. Shank teaches that application 140 must use an interface specified by the text-to-speech service's ORB vendor (Shank, column 3, line 65-column 4, line 6). Thus, Shank's text-to-speech service is not a selectable message format for communicating between a manager and a managed object. Shank clearly does not describe a manager being able to select a format for message delivery that comprises text.

Claim 40:

In regard to claim 40, Appellants have argued that Shank does not anticipate a gateway comprising a request gateway which is configured to deliver messages generated by the one or more managers to the one or more managed objects, wherein the messages comprises requests and wherein the requests comprise a query for information concerning one of the managed objects. The portions of Shank cited by the Examiner (column 2, lines 64-67; column 7, lines 43-46; and column 7, line 66 through column 8, line 6) refer to application 140 invoking functions of the telephony and media services. These teachings have nothing to do with a query for information concerning a managed object. The concepts of managers and managed objects are well understood in the art of managed networks. Managers and managed objects have a well-known relationship in managed networks. Shank does not pertain to interactions between managers and managed objects, as these entities are understood in the art. In contrast, Shank only discusses the client-server interactions between application 140 and server 110. In other words, Shank only discusses providing telephony and media services through a server to a client application. Shank does not discuss managing managed network objects. Furthermore, nowhere does Shank teach a query for information concerning a managed object.

In the Examiner's Answer the Examiner cites item 230 of FIG. 2 and column 5, lines 13-26 of Shank and argues that Shank discloses requesting the available resources by the user based on the interface type and properties. However, the Examiner's characterization of the cited passage is incorrect. The cited passage describes Shank's resource administrator and how resources may register with the resource administrator by providing information including type and, optionally, any attributes or properties that distinguish it from other resources. Shank states that resources "can query resource administrator 236 for the existence of resources by interface type and an optional list of properties" (Shank, column 5, lines 20-22). Thus, the cited passage does not disclose a user requesting available resources, as the Examiner asserts. Instead, it teaches that resources may query a resource administrator regarding other resources. However, a resource making a query regarding other resources is not equivalent to a message generated by a manager to a managed object where the message includes a request that comprises a query for information concerning a managed object. Shank's resources are clearly not managers, and Shank's resource administrator is not a managed object. Thus any query a resource makes to the resource administrator cannot be considered a message generated by a manager to a managed object. The Examiner has clearly misinterpreted the teachings of Shank.

Claim 41:

Regarding claim 41, Shank does not disclose that the requests comprise a command to set one or more parameters of one of the managed objects. The Examiner cites column 17, lines 53-66 where Shank describes parameters for a specific function (Play) of the Player media service interface, but does not mention a command to set parameters for a managed object. A parameter to a specific service method invocation is very different from a command to set one or more parameters of a managed object and Shank does not mention such a command, just that parameters may be used when invoking specific service method invocations.

In the Examiner's Answer the Examiner cites column 10, lines 55-61 of Shank and refers to Shank's setParametersAsync used to set session parameters. The Examiner further states, "calling the objects satisfy the request criteria." However, the setParameterAsync method is a part of Shank's Session IDL API that "provides a logical binding between application 140 and media server 200" (Shank, column 8, lines 37-40). The setParameterAsync method is not a message generated by a manager to a managed object, but instead is between an application and a media server. Additionally, Shank's setParameterAsync method does not set parameters of a managed object. Instead, setParameterAsync sets session parameters, which are not parameters of a data resource, which the Examiner equates to the managed objects of Appellants' claims.

Claim 43:

In regard to claim 43, Shank fails to anticipate, wherein requests are converted from the interface definition language to a platform-specific format prior to delivery to the managed objects. The Examiner refers to col. 5, lines 39-50, of Shank. The Examiner's cited portion of Shank discusses examples of media and telephony services but teaches nothing about converting requests from the interface definition language to a platform-specific format prior to delivery to the managed objects. In fact, nowhere does Shank teach or even suggest that requests are converted. Thus, appellants can see no basis for the Examiner's contention regarding converting requests and assert that the Examiner is merely speculating about Shank's system including converting requests.

In the Examiner's Answer, the Examiner refers to Shank's teachings regarding the communication with different objects by different protocols "based on an industry standard" and further argues, "ASN1 can be implemented in Shank's system." Appellants fail to find any relevance to the Examiner's reference to ASN1. Shank does not mention ASN1 at all and certainly does not teach or suggest the use of ASN1. Furthermore, even if ASN1 were to be implemented in Shank's system, that would not require that Shank's system include the translation of request messages. The Examiner seems to be implying that it would be obvious to modify Shank's system to include the

translation of request messages; however, a rejection based on such modification is clearly improper in a § 102(e) anticipation rejection.

The Examiner further argues that Shank teaches "converting requests before delivering them to objects when the user and server [execute] in different process[es]" and cites column 4, lines 35-40 of Shank. However, this portion of Shank discusses the marshalling of a method invocation in order to communicate between a client in one process and a server in a different process. Marshalling of parameters for method invocations does not involve the translation of any request messages. Shank is not discussing translating request messages at all, but rather Shank is discussing parameters used in inter-process communication.

Shank clearly does not anticipate wherein requests are converted from an interface definition language to a platform-specific format prior to delivery to the managed objects.

Third Ground of Rejection:

Claim 3:

Shank fails to teach or suggest wherein the selected format comprises Abstract Syntax Notation One (ASN1). The Examiner has not provided any prior art reference or specific finding that provides a motivation to use ASN.1 in Shank in any way. The Examiner states only that such modifications would be obvious "for fulfilling the system requirements." However, there are no system requirements taught in Shank that would require or even suggest selecting ASN.1 as a message format. The Examiner's stated motivation for modifying Shank ("fulfilling the system requirement") amounts to nothing more than an attempt to build Appellants' invention through hindsight analysis and thus is clearly improper.

In response, the Examiner, in the Examiner's Answer, refers to Shank's teachings regarding the communication with different objects by different protocols "based on an

industry standard" and further argues, "ASN1 can be implemented in Shank's system." Appellants fail to find any relevance to the Examiner's reference to ASN1. Shank does not mention ASN1 at all and certainly does not teach or suggest the use of ASN1. Furthermore, even if ASN1 were to be implemented in Shank's system, that would not require that Shank's system include the translation of request messages. The Examiner seems to be implying that it would be obvious to modify Shank's system to include the translation of request messages; however, a rejection based on such modification is clearly improper in a § 102(e) anticipation rejection.

Claim 12:

Shank fails to teach or suggest wherein the requests are converted from the interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed objects. The Examiner has not cited any passage of Shank (or any other prior art reference) that teaches or suggests converting requests from IDL to PMI format. Nor has the Examiner presented any motivation to modify Shank to convert requests from interface definition language to a PMI format prior to delivery to the managed objects, as the Examiner contends. Nor is there any teaching in Shank that would require or even suggest converting requests from the interface definition language to a PMI format (or any other format) prior to delivery to the managed objects. As with the rejection of claim 3, discussed above, The Examiner's stated motivation for modifying Shank ("fulfilling the system requirement") amounts to nothing more than an attempt to build the appellants invention through hindsight analysis and is clearly improper.

In the Examiner's Answer, as well as in the Response to Arguments section of the Final Action, the Examiner refers to Shank's teachings regarding the communication with different objects by different protocols "based on an industry standard" and further argues, "ASN1 can be implemented in Shank's system." Appellants fail to find any relevance to the Examiner's reference to ASN1. Shank does not mention ASN1 at all and certainly does not teach or suggest the use of ASN1. Furthermore, even if ASN1 were to

be implemented in Shank's system, that would not require that Shank's system include the translation of request messages. The Examiner seems to be implying that it would be obvious to modify Shank's system to include the translation of request messages; however, a rejection based on such modification is clearly improper in a § 102(e) anticipation rejection.

The Examiner further argues that Shank teaches "converting requests before delivering them to objects when the user and server [execute] in different process[es]" and cites column 4, lines 35-40 of Shank. However, the portion of Shank cited by the Examiner discusses the marshalling of a method invocation in order to communicate between a client in one process and a server in a different process. Marshalling of parameters for method invocations does not involve the translation of any request messages. Shank is not discussing translating request messages at all, but rather Shank is discussing parameters used in inter-process communication. Shank clearly does not anticipate that requests are converted from an interface definition language to a platform-specific format prior to delivery to the managed objects.

The Examiner further argues that the format of converted requests in Shank would be based on the particular receiver, and thus, "they could be any industry standard format, such as Portable Management Interface (PMI) format." As noted above, the Examiner is attempting to reject claim 13 based on features that might or could be implemented in Shank's system, not based upon what is actually disclosed by Shank, which is clearly improper in a rejection based on anticipation (i.e. § 102).

Additionally, Appellants' remarks above regarding the § 102(e) rejection of claim 13 in view of Shank apply here.

Claim 18:

Shank fails to teach or suggest wherein the selected format comprises Abstract Syntax Notation One (ASN1). The Examiner has not provided any prior art reference or

specific finding that provides a motivation to use ASN.1 in Shank in any way. The Examiner states only that such modifications would be obvious "for fulfilling the system requirements." However, there are no system requirements taught in Shank that would require or even suggest selecting ASN.1 as a message format. The Examiner's stated motivation for modifying Shank ("fulfilling the system requirement") amounts to nothing more than an attempt to build the appellants invention through hindsight analysis and thus is clearly improper.

In response, the Examiner, in the Examiner's Answer, refers to Shank's teachings regarding the communication with different objects by different protocols "based on an industry standard" and further argues, "ASN1 can be implemented in Shank's system." Appellants fail to find any relevance to the Examiner's reference to ASN1. Shank does not mention ASN1 at all and certainly does not teach or suggest the use of ASN1. Furthermore, even if ASN1 were to be implemented in Shank's system, that would not require that Shank's system include the translation of request messages. The Examiner seems to be implying that it would be obvious to modify Shank's system to include the translation of request messages; however, a rejection based on such modification is clearly improper in a § 102(e) anticipation rejection.

Claim 27:

Shank fails to teach or suggest wherein the requests are converted from the interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed objects. The Examiner has not cited any passage of Shank (or any other prior art reference) that teaches or suggests converting requests from IDL to PMI format. Nor has the Examiner presented any motivation to modify Shank to convert requests from interface definition language to a PMI format prior to delivery to the managed objects, as the Examiner contends. Nor is there any teaching in Shank that would require or even suggest converting requests from the interface definition language to a PMI format (or any other format) prior to delivery to the managed objects. As with the rejection of claim 3, discussed above, The Examiner's stated motivation for

modifying Shank ("fulfilling the system requirement") amounts to nothing more than an attempt to build the appellants invention through hindsight analysis and is clearly improper.

In the Examiner's Answer, as well as in the Response to Arguments section of the Final Action, the Examiner refers to Shank's teachings regarding the communication with different objects by different protocols "based on an industry standard" and further argues, "ASN1 can be implemented in Shank's system." Appellants fail to find any relevance to the Examiner's reference to ASN1. Shank does not mention ASN1 at all and certainly does not teach or suggest the use of ASN1. Furthermore, even if ASN1 were to be implemented in Shank's system, that would not require that Shank's system include the translation of request messages. The Examiner seems to be implying that it would be obvious to modify Shank's system to include the translation of request messages; however, a rejection based on such modification is clearly improper in a § 102(e) anticipation rejection.

The Examiner further argues that Shank teaches "converting requests before delivering them to objects when the user and server [execute] in different process[es]" and cites column 4, lines 35-40 of Shank. However, the portion of Shank cited by the Examiner discusses the marshalling of a method invocation in order to communicate between a client in one process and a server in a different process. Marshalling of parameters for method invocations does not involve the translation of any request messages. Shank is not discussing translating request messages at all, but rather Shank is discussing parameters used in inter-process communication. Shank clearly does not anticipate that requests are converted from an interface definition language to a platform-specific format prior to delivery to the managed objects.

The Examiner further argues that the format of converted requests in Shank would be based on the particular receiver, and thus, "they could be any industry standard format, such as Portable Management Interface (PMI) format." As noted above, the Examiner is attempting to reject claim 13 based on features that might or could be implemented in

Shank's system, not based upon what is actually disclosed by Shank, which is clearly improper in a rejection based on anticipation (i.e. § 102).

Additionally, Appellants' remarks above regarding the § 102(e) rejection of claim 28 apply here.

Claim 33:

Shank fails to teach or suggest wherein the selected format comprises Abstract Syntax Notation One (ASN1). The Examiner has not provided any prior art reference or specific finding that provides a motivation to use ASN.1 in Shank in any way. The Examiner states only that such modifications would be obvious "for fulfilling the system requirements." However, there are no system requirements taught in Shank that would require or even suggest selecting ASN.1 as a message format. The Examiner's stated motivation for modifying Shank ("fulfilling the system requirement") amounts to nothing more than an attempt to build the appellants invention through hindsight analysis and thus is clearly improper.

In response, the Examiner, in the Examiner's Answer, refers to Shank's teachings regarding the communication with different objects by different protocols "based on an industry standard" and further argues, "ASN1 can be implemented in Shank's system." Appellants fail to find any relevance to the Examiner's reference to ASN1. Shank does not mention ASN1 at all and certainly does not teach or suggest the use of ASN1. Furthermore, even if ASN1 were to be implemented in Shank's system, that would not require that Shank's system include the translation of request messages. The Examiner seems to be implying that it would be obvious to modify Shank's system to include the translation of request messages; however, a rejection based on such modification is clearly improper in a § 102(e) anticipation rejection.

Claim 42:

Further regarding claim 42, Shank fails to disclose wherein the requests are converted from the interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed objects. The Examiner provided any prior art reference or specific finding that provides a motivation to modify Shank to convert requests from the interface definition language to a PMI format prior to delivery to the managed objects, as the Examiner contends. Nor is there any teaching in Shank that would require or even suggest converting requests from the interface definition language to a PMI format prior to delivery to the managed objects. As with the rejection of claim 3, discussed above, The Examiner's stated motivation for modifying Shank ("fulfilling the system requirement") amounts to nothing more than an attempt to build the appellants invention through hindsight analysis and is clearly improper.

In the Examiner's Answer, as well as in the Response to Arguments section of the Final Action, the Examiner refers to Shank's teachings regarding the communication with different objects by different protocols "based on an industry standard" and further argues, "ASN1 can be implemented in Shank's system." Appellants fail to find any relevance to the Examiner's reference to ASN1. Shank does not mention ASN1 at all and certainly does not teach or suggest the use of ASN1. Furthermore, even if ASN1 were to be implemented in Shank's system, that would not require that Shank's system include the translation of request messages. The Examiner seems to be implying that it would be obvious to modify Shank's system to include the translation of request messages; however, a rejection based on such modification is clearly improper in a § 102(e) anticipation rejection.

The Examiner further argues that Shank teaches "converting requests before delivering them to objects when the user and server [execute] in different process[es]" and cites column 4, lines 35-40 of Shank. However, the portion of Shank cited by the Examiner discusses the marshalling of a method invocation in order to communicate between a client in one process and a server in a different process. Marshalling of

parameters for method invocations does not involve the translation of any request messages. Shank is not discussing translating request messages at all, but rather Shank is discussing parameters used in inter-process communication. Shank clearly does not anticipate that requests are converted from an interface definition language to a platform-specific format prior to delivery to the managed objects.

The Examiner further argues that the format of converted requests in Shank would be based on the particular receiver, and thus, "they could be any industry standard format, such as Portable Management Interface (PMI) format." As noted above, the Examiner is attempting to reject claim 13 based on features that might or could be implemented in Shank's system, not based upon what is actually disclosed by Shank, which is clearly improper in a rejection based on anticipation (i.e. § 102).

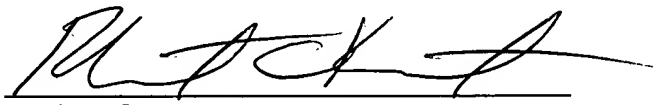
Additionally, Appellants' remarks above regarding the § 102(e) rejection of claim 43 in view of Shank apply here.

CONCLUSION

For the foregoing reasons submitted in the Appeal Brief and this Reply Brief, it is submitted that the Examiner's rejections of claims 1-45 are erroneous. Reversal of the Examiner's decision is respectfully requested.

The Commissioner is authorized to charge any fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-61100/RCK. This Reply Brief is submitted with a return receipt postcard.

Respectfully submitted,



Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: November 3, 2005